Adaptive control of robotic manipulators with unified motion constraints

Article (Accepted Version)

Adaptive Control of Robotic Manipulators with Unified Motion Constraints

Mingming Li, Yanan Li, *Member, IEEE* Shuzhi Sam Ge, *Fellow, IEEE* Tong Heng Lee, *Senior Member, IEEE*

***Abstract*—In this paper, we present an adaptive control of robotic manipulators with parametric uncertainties and motion constraints. Position and velocity constraints are considered and they are unified and converted into the constraint of the nominal input. An adaptive neural network control is developed to achieve trajectory tracking, while the problems of motion constraints are addressed by considering the saturation effect of the nominal input. The uniform boundedness of all closed-loop signals is verified through Lyapunov analysis. Simulation and experiment results on a 2 DOF robotic manipulator demonstrate the effectiveness of the proposed method.**

***Index Terms*—robotic manipulator; adaptive control; unified motion constraints; input saturation; neural network approximation**

I. Introduction

In robotic applications with unstructured environments, especially those related to human-robot interaction [1], [2], [3], [4], [5], robots must be subject to certain motion constraints, which typically include position and velocity constraints. In the example of teleoperated surgical robots, they must be maneuvered in constrained task spaces, often through a narrow entry portal into the patient's body [6]. In another example of human-robot collaboration, a robot shares a common workspace with a human, so its velocity must be constrained within a predefined bound to guarantee the safety of both the robot and the human. An impact test is conducted in [7] to evaluate endangerment to human beings caused by collision with robots moving at different velocities. Results have shown that the head injury criterion (HIC), which is used to quantify the injury level of human beings, significantly increases as the impact velocity of robots becomes larger.

In the literature, research effort has been made on control of robotic manipulators with motion constraints. A bounded control for set-point regulation problem of robotic manipulators with joint velocity constraints is proposed in [8]. In [9], a trajectory tracking control of robotic manipulators is achieved on a surface with joint velocity constraints. A unified quadratic programming formulation based dynamical system approach is proposed in [10] to solve the joint torque optimization problem of physically constrained redundant manipulators, but it requires complete knowledge of the system dynamics. In [11], [12], adaptive control of robotic manipulators with position (output) constraints is developed by employing the barrier Lyapunov function (BLF), in the presence of parametric uncertainties and external disturbances. In [13], optimal control of robotic manipulators with joint position constraints is achieved using adaptive dynamic programming. Researches of task space region control are proposed in [14], [15], [16] and adaptive control of robot subject to position constraints is achieved in [17] by defining the objective functions in the form of a set of inequalities. However, only position constraints are considered in the designed region control of [17] as these objective functions (the inequalities) are constructed with respect to only the robot's task space positions.

Alternatively, since a robotic manipulator can be considered as a nonlinear system subject to some physical properties [18], nonlinear control with consideration of state constraints may be applied [19], [20], [21]. Neural network based predictive control is proposed in [22], [23] with system identification implemented using historical data and off-line training while predictive control with online estimation proposed in [24], [25], [26] is only applicable to single-input-single-output (SISO) system. Recently, a robust adaptive neural tracking control with integral BLF [27] is proposed in [28] with integral BLFs for a class of SISO strict-feedback nonlinear systems under state constraints while in [29], adaptive control subject to full state constraints is achieved for a more general SISO pure-feedback nonlinear system, with simulation results on a single-link robot. To tackle the uncertainties in multi-input-multi-output (MIMO) systems like robots, neural networks [30], [31] or fuzzy logic systems (FLS) [32], [33], [34] are employed as online model-free approximators. In [35], an adaptive fault tolerant control is derived for a class of input and state constrained MIMO nonlinear systems, where the state constraints are formulated as a constraint of the state vector's norm. Therefore, it implies that the state constraints have to be symmetric. On the other hand, in [36], [37], prescribed performance adaptive neural network control is developed to constrain the error signals within a prescribed region, which is shaped by a performance function. This method may be extended to handle the problem of motion constraints, but it is not straightforward to define the corresponding performance function which is related to the tracking error and is required to be exponentially decaying.

In this paper, an adaptive neural network control is proposed for robotic manipulators with unknown dynamics and motion constraints of position and velocity. This method is different from previous methods since it is able to directly handle various types of motion constraints that are independently and explicitly defined. Specifically, a unified framework is established to convert motion constraints into an input saturation

M. Li, S. S. Ge and T. H. Lee are with the Department of Electrical and Computer Engineering, and the Social Robotics Lab, Interactive and Digital Media Institute (IDMI), National University of Singapore, Singapore 117576 (e-mail: li_mingming@u.nus.edu ; samge@nus.edu.sg; eleleeth@nus.edu.sg).

Y. Li is with Department of Bioengineering, Imperial College London, London, UK SW7 2AZ (e-mail: hit.li.yn@gmail.com.)

problem with uncertainties incorporated in both the nominal control input and saturation law, which is different from the input saturation problem in [38]. By using the GL matrix and operator [39], an adaptive neural network approximator is designed to address the issue of unknown system dynamics. By Lyapunov analysis, the boundedness of all closed-loop signals is shown to be guaranteed while the motion constraints are not violated. Simulations and experiments are conducted to illustrate the efficacy of the proposed method.

The remainders of this paper are organized as follows. In Section II, the model of a robotic manipulator with motion constraints is presented, a general method is introduced to convert motion constraints into the saturation of the control input, and some useful preliminaries are given. In Section III, the adaptive control design is detailed with rigorous Lyapunov stability analysis. Simulation and experiment results are presented in Sections IV and V to demonstrate the effectiveness of the proposed method, followed by some concluding remarks in Section VI.

II. Problem Formulation

A. Kinematics and Dynamics

The dynamics of an n degree-of-freedom (DOF) robotic manipulator can be described as a MIMO nonlinear system as follows [18]:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau \tag{1}$$

where $q \in \mathbb{R}^n$ is the joint position, $M(q) \in \mathbb{R}^{n\times n}$ is the inertia matrix, $C(q,\dot{q})\dot{q} \in \mathbb{R}^n$ is the Coriolis and centrifugal force, $G(q) \in \mathbb{R}^n$ is the gravitational force, and $\tau \in \mathbb{R}^n$ is the input joint torque. The joint space dynamics (1) are transformed into the task space dynamics

$$M_\eta(\eta)\ddot{\eta} + C_\eta(\eta,\dot{\eta})\dot{\eta} + G_\eta(\eta) = u \tag{2}$$

via the forward kinematics

$$\eta = \Omega(q), \qquad \dot{\eta} = \frac{\partial \Omega}{\partial q}\dot{q} =: J(q)\dot{q} \tag{3}$$

where $\eta = [\eta_1, \eta_2, \cdots, \eta_m]^{\mathrm{T}}$ is a vector of task variables, M_η is the inertia matrix, C_η is the Coriolis and centrifugal force, G_η is the gravitational force, and u is the control input in the task space given as

$$\begin{aligned} M_\eta &= J^{-\mathrm{T}} M J^{-1}, G_\eta = J^{-\mathrm{T}} G, \\ C_\eta &= J^{-\mathrm{T}}(C - MJ^{-1}\dot{J})J^{-1}, u = J^{-\mathrm{T}}\tau \end{aligned} \tag{4}$$

For simplicity, only the non-redundant $(m = n)$ non-singular manipulators are considered in this paper. Let $x_1(t) = \eta, x_2(t) = \dot{\eta}$, we have the dynamics of the robotic manipulator in the state-space form, as below

$$\begin{cases} \dot{x}_1 = x_2, \\ \dot{x}_2 = M_\eta^{-1}(x_1)(u - C_\eta(x_1,x_2)x_2 - G_\eta(x_1)), \\ y = x_1 \end{cases} \tag{5}$$

Note that $M_\eta^{-1}(x_1), C_\eta(x_1,x_2)$ and $G_\eta(x_1))$ are all unknown matrices. The control objective is to make the position of the end-effector $y(t)$ track the desired trajectory $y_{\mathrm{d}}(t)$. In addition, it is expected to keep all closed-loop signals bounded and prevent the violation of motion constraints, which will be introduced in the following.

Assumption 1: [40] For arbitrary $t > 0$, there exist $b_1 > 0$ and $b_2 > 0$ such that $\|\dot{y}_{\mathrm{d}}(t)\| \le b_1$ and $\|\ddot{y}_{\mathrm{d}}(t)\| \le b_2$.

B. Motion Constraints

A motion constraint of the end-effector is a kinematic/dynamic constraint related to the variable that describes the end-effector's motion status. In this paper, we consider two typical motion constraints, i.e., position and velocity constraints.

For the robotic manipulator (5), let y_i, y_{di}, x_{1i} and x_{2i} denote the i-th element of the vectors y, y_d, x_1 and x_2 for $i = 1, \cdots, n$, respectively. Motion constraints are defined as

$$p_i^- \le y_i \le p_i^+, \text{ and } v_i^- \le \dot{y}_i \le v_i^+ \tag{6}$$

for $i = 1, 2, \cdots, n$, where p_i^-, v_i^-, p_i^+ and v_i^+ are known constant or time-varying limits.

Remark 1: Note that the constraints investigated in this paper are given in the form of two-sided inequalities. Compared with those considered in [28] and [35], which are described as bounds of the absolute value/norm of a state variable, respectively, the constraints formulated in (6) are asymmetric and they can be assigned independently as the upper and lower bounds of the state variable. Therefore, it can account for more general task space constraints in practice and offer greater flexibility during control design.

Assumption 2: The initial position and velocity of the end-effector satisfy $p_i^- \le y_i(0) \le p_i^+$ and $v_i^- \le \dot{y}_i(0) \le v_i^+$ for $i = 1, 2, \cdots, n$.

Based on the above assumption, the two-sided inequalities of the motion constraints are converted following the procedures below. Choose positive scalars k_{1i} ,k_{2i}, and k_{3i}, and consider the scalar $\dot{y}_i$ regulated as

$$-k_{1i}(y_i - p_i^-) \le \dot{y}_i \le -k_{1i}(y_i - p_i^+) \tag{7}$$

Considering the following inequalities

$$\begin{aligned} \dot{y}_i \ge -k_{1i}(y_i - p_i^-) \ge 0 \quad &\text{if} \quad y_i \le p_i^-, \text{ and} \\ \dot{y}_i \le -k_{1i}(y_i - p_i^+) \le 0 \quad &\text{if} \quad y_i \ge p_i^+ \end{aligned} \tag{8}$$

we find that y_i cannot transgress the constraints when $\dot{y}_{i\min} = -k_{1i}(y_i - p_i^-)$ and $\dot{y}_{i\max} = -k_{1i}(y_i - p_i^+)$. Similarly, we can guarantee $\dot{y}_i \in [\dot{y}_{i\min}, \dot{y}_{i\max}]$ if $\ddot{y}_i$ satisfies

$$-k_{2i}(\dot{y}_i - \dot{y}_{i\min}) \le \ddot{y}_i \le -k_{2i}(\dot{y}_i - \dot{y}_{i\max}) \tag{9}$$

Thus, according to (5), we have

$$h_{\mathrm{p}i}^- \le \dot{x}_{2i} \le h_{\mathrm{p}i}^+ \tag{10}$$

where $h_{\mathrm{p}i}^- = -k_{2i}(x_{2i} + k_{1i}(x_{1i} - p_i^-))$ and $h_{\mathrm{p}i}^+ = -k_{2i}(x_{2i} + k_{1i}(x_{1i} - p_i^+))$.

Consider the velocity constraint of the end-effector defined as $v_i^- \le x_{2i} \le v_i^+$. Similarly to the process of converting the position constraint, we have

$$-k_{3i}(x_{2i} - v_i^-) \le \dot{x}_{2i} \le -k_{3i}(x_{2i} - v_i^+) \tag{11}$$

Thus, $\dot{x}_{2i}$ is constrained as

$$h_{\mathrm{v}i}^- \leq \dot{x}_{2i} \leq h_{\mathrm{v}i}^+ \tag{12}$$

where $h_{\mathrm{v}i}^- = -k_{3i}(x_{2i} - v_i^-)$ and $h_{\mathrm{v}i}^+ = -k_{3i}(x_{2i} - v_i^+)$. Thus, x_{2i} will not violate the velocity constraint $v_i^- \leq x_{2i} \leq v_i^+$ if $\dot{y}_i(0) = x_{2i}(0) \in [v_i^-, v_i^+]$.

As both (10) and (12) are with respect to $\dot{x}_2$, they can be combined as a unified constraint as below

$$h_i^- \leq \dot{x}_{2i} \leq h_i^+ \tag{13}$$

where $h_i^- = \max\{h_{\mathrm{p}i}^-, h_{\mathrm{v}i}^-\}$ and $h_i^+ = \min\{h_{\mathrm{p}i}^+, h_{\mathrm{v}i}^+\}$.

Remark 2: In this way, the position and velocity constraints have been converted into a unified one as (13). In fact, it is clear that one can even incorporate the robot's acceleration constraints $a_i^- \leq \dot{x}_{2i} \leq a_i^+$ into (13) if necessary, which requires no change in control design as all motion constraints are compactly represented as (13) and our control is derived solely based on it. In addition, such a conversion process can also be applied to constraints in the joint space when the robotic manipulator is subject to different types of joint limits and a joint space controller is to be designed. Comparatively, one may extend region control with the unified objective bound in [17] to handle velocity or even acceleration constraints by adding the corresponding inequalities of the task space velocities and accelerations into the objective functions. However, considering these additional constraints with the method in [17] will require the signals $\ddot{q}$ and $\dddot{q}$ in the resulted controller as it is derived by taking the time-derivatives of the objective functions. Considering the fact that the usage of $\ddot{q}$ or even $\dddot{q}$ in control design is generally not desired due to the difficulties in obtaining their accurate measurements [18], the practicability of this region control method for motion constraints investigated in this paper may be limited.

C. Preliminaries: Neural Network

It has been shown that the radial basis function neural network (RBFNN) can approximate an arbitrary continuous function $a(Z)$ over a compact set $\Omega_Z \subset \mathbb{R}^{n_Z}$ to any accuracy [41] as below

$$a(Z) = w^{*\mathrm{T}}\phi(Z) + e_Z, \forall Z \in \Omega_Z \tag{14}$$

where $Z \in \mathbb{R}^{n_Z}$ is the input vector, $w^* \in \mathbb{R}^{n_w}$ are the optimal constant weights with n_w being the number of nodes, $e_Z \in \mathbb{R}$ is the functional approximation error, and $\phi(Z) = [\phi^{(1)}(Z), \phi^{(2)}(Z), \cdots, \phi^{(n_w)}(Z)] \in \mathbb{R}^{n_w}$ are vectors of Gaussian functions as below

$$\phi^{(i)}(Z) = \exp\left(\frac{-(Z-\mu^{(i)})^{\mathrm{T}}(Z-\mu^{(i)})}{\sigma_i^{(i)2}}\right) \tag{15}$$

with $\mu^{(i)}$ being the center of the Gaussian function and $\sigma^{(i)}$ being the variance.

According to [42], there exist optimal weights w^* such that $|e_Z| \leq e_Z^*$ with $e_Z^* \geq 0$, which can be made arbitrary small provided that n_w is sufficiently large. Thus, the optimal weights w^* are defined such that e_Z is minimized for all $Z \subset \Omega_Z$, i.e.,

$$w^* := \arg\min_{w \in \mathbb{R}^{n_w}} \{\sup_{Z \subset \Omega_Z} |a(Z) - w^{\mathrm{T}}\phi(Z)|\}. \tag{16}$$

Then, an approximation of $a(Z)$ can be constructed as

$$\hat{a}(Z) = \hat{w}^{\mathrm{T}}\phi(Z) \tag{17}$$

where $\hat{a}(Z)$ is the approximation of $a(Z)$ and $\hat{w} \in \mathbb{R}^{n_w}$ are the estimates of the corresponding optimal weights w^* defined in (16).

By employing RBFNNs to approximate each of its element, a matrix function $A(Z) \in \mathbb{R}^{n_1 \times n_2}$ is approximated as $\hat{A}(Z)$,. Following the convention of GL matrices and GL operator for vectors and matrices [39], the expression of $\hat{A}(Z)$ is given as

$$\begin{aligned}
\hat{A}(Z) &= \begin{bmatrix} \hat{a}_{11}(Z) & \hat{a}_{12}(Z) & \cdots & \hat{a}_{1n_2}(Z) \\ \hat{a}_{21}(Z) & \hat{a}_{22}(Z) & \cdots & \hat{a}_{2n_2}(Z) \\ \vdots & \vdots & \cdots & \vdots \\ \hat{a}_{n_1 1}(Z) & \hat{a}_{n_1 2}(Z) & \cdots & \hat{a}_{n_1 n_2}(Z) \end{bmatrix} \\
&= \{\hat{W}\}^{\mathrm{T}} \bullet \{\Phi\} \\
&= \begin{bmatrix} \hat{w}_{11}^{\mathrm{T}}\phi_{11} & \hat{w}_{12}^{\mathrm{T}}\phi_{12} & \cdots & \hat{w}_{1n}^{\mathrm{T}}\phi_{1n_2} \\ \hat{w}_{21}^{\mathrm{T}}\phi_{21} & \hat{w}_{22}^{\mathrm{T}}\phi_{22} & \cdots & \hat{w}_{2n}^{\mathrm{T}}\phi_{2n_2} \\ \vdots & \vdots & \cdots & \vdots \\ \hat{w}_{n_1 1}^{\mathrm{T}}\phi_{n_1 1} & \hat{w}_{n_1 2}^{\mathrm{T}}\phi_{n_1 2} & \cdots & \hat{w}_{n_1 n_2}^{\mathrm{T}}\phi_{n_1 n_2} \end{bmatrix}
\end{aligned}$$

where $\{W\}, \{\Phi\}$ are the GL matrices and $\bullet$ is the GL operator.

Remark 3: In this paper, RBF neural networks are employed due to its capabilities of approximating any unstructured smooth nonlinear functions to arbitrary accuracy over a compact set. In fact, one can use other online approximation method instead, such as fuzzy logic systems [43], [44], [45], [32]. The online approximator can be further reduced to a regressor function by assuming that the uncertainties are structured and are linear in parameters [46], which requires a set of model-specific basis functions.

III. Control Design

To incorporate the constraint (13) into the actual control input u, the state equation in (5) is rewritten as

$$\dot{x}_2 = U + P(x_1, x_2) \tag{18}$$

where $U = M_\eta^{-1}(x_1)u \in \mathbb{R}^n$ is the nominal control input and $P(x_1, x_2) = -M_\eta^{-1}(x_1)(C_\eta(x_1, x_2)x_2 + G_\eta(x_1)) \in \mathbb{R}^n$. Let U_i and $P_i(x_1, x_2)$ denote the i-th element of the vector U and $P(x_1, x_2)$, respectively. In order to guarantee (13), U_i has to be saturated as

$$h_i^- - P_i(x_1, x_2) \leq U_i \leq h_i^+ - P_i(x_1, x_2) \tag{19}$$

Note that both U and $P(x_1, x_2)$ incorporate unknown components, i.e., $M_\eta^{-1}(x_1)$, $C_\eta(x_1, x_2)$, and $G_\eta(x_1)$. Thus, we construct RBFNNs to express M_η^{-1} and P as below

$$M_\eta^{-1} = \{W_M^*\}^{\mathrm{T}} \bullet \{\Phi_M(x_1)\} + E_M \tag{20}$$

$$P = \{W_P^*\}^{\mathrm{T}} \bullet \{\Phi_P(x_1, x_2)\} + E_P \tag{21}$$

where $\{W_M^*\}, \{W_P^*\}, \{\Phi_M(x_1)\}$, and $\{\Phi_P(x_1, x_2)\}$ are GL matrices formed by optimal neural network weight vectors $W_{Mij}^* \in \mathbb{R}^{n_{W_M}}$ and $W_{Pi}^* \in \mathbb{R}^{n_{W_P}}$, and basis function vectors $\phi_{Mij} \in \mathbb{R}^{n_{W_M}}$ and $\phi_{Pi} \in \mathbb{R}^{n_{W_P}}$, respectively. $E_M \in \mathbb{R}^{n \times n}$ and $E_P \in \mathbb{R}^n$ are formed by approximation errors e_{Mij} and e_{Pi}, respectively. Let $\bar{E}_M$ and $\bar{E}_P$ denote the matrices formed by $\bar{e}_{Mij}$ and $\bar{e}_{Pi}$, which are the corresponding upper bounds

of e_{Mij} and e_{Pi} used to deal with the approximation errors later. Then, unknown matrices M_η^{-1} and P are estimated as

$$\hat{M}_\eta^{-1} = \{\hat{W}_M\}^{\mathrm{T}} \bullet \{\Phi_M(x_1)\} \quad (22)$$

$$\hat{P} = \{\hat{W}_P\}^{\mathrm{T}} \bullet \{\Phi_P(x_1, x_2)\} \quad (23)$$

The nominal control input in (18) is estimated as

$$\hat{U} = (\hat{M}_\eta^{-1} - \delta)u = (\{\hat{W}_M\}^{\mathrm{T}} \bullet \{\Phi_M(x_1)\} - \delta)u \quad (24)$$

where δ will be used later to deal with the presence of functional approximation errors E_M. Similarly to (19), with the estimated terms (23) and (24), we can obtain the constraints of the estimated nominal control input as follows:

$$h_i^- - \hat{P}_i(x_1, x_2) \le \hat{U}_i \le h_i^+ - \hat{P}_i(x_1, x_2) \quad (25)$$

Hereinafter, we will show that the estimated weights in (23) and (24) will exponentially converge at a rate that can be controlled by properly choosing the design parameters. In addition, the functional approximation errors can be made arbitrarily small by using sufficiently large numbers of neurons [41]. Therefore, (25) is valid to prevent the violation of motion constraints.

Let $U_0 \in \mathbb{R}^n$ and U_{0i} denote the unsaturated nominal control and its i-th element, respectively. From (25), the relationship between $\hat{U}_i$ and U_{0i} is given as

$$\hat{U}_i = \begin{cases} h_i^- - \hat{P}_i(x_1, x_2), & \text{if } U_{0i} < h_i^- - \hat{P}_i(x_1, x_2) \\ h_i^+ - \hat{P}_i(x_1, x_2), & \text{if } U_{0i} > h_i^+ - \hat{P}_i(x_1, x_2) \\ U_{0i}, & \text{otherwise} \end{cases} \quad (26)$$

To achieve the control objective, one needs to properly design U_0, which will be introduced later. Then, according to (24) and (26) and taking the inverse of $\{\hat{W}_M\}^{\mathrm{T}} \bullet \{\Phi_M(x_1)\} - \delta$, we have

$$u' = (\{\hat{W}_M\}^{\mathrm{T}} \bullet \{\Phi_M(x_1)\} - \delta)^{-1}\hat{U} \quad (27)$$

Note that, in (27), the estimated term $\{\hat{W}_M\}^{\mathrm{T}} \bullet \{\Phi_M(x_1)\} - \delta$ may be ill-conditioned or even singular. Thus, we perform singular value decomposition [47] with this term and have

$$\{\hat{W}_M\}^{\mathrm{T}} \bullet \{\Phi_M(x_1)\} - \delta = U_s \Sigma_s V_s^{\mathrm{T}} \quad (28)$$

where U_s and $V_s \in \mathbb{R}^{n\times n}$ are two orthogonal matrices, and $\Sigma_s \in \mathbb{R}^{n\times n}$ is a diagonal matrix formed by the singular values σ_{si} of the matrix $\{\hat{W}_M\}^{\mathrm{T}} \bullet \{\Phi_M(x_1)\} - \delta$. Then, instead of forming the pseudoinverse of $\{\hat{W}_M\}^{\mathrm{T}} \bullet \{\Phi_M(x_1)\} - \delta$ by directly replacing every nonzero diagonal entries with their reciprocals, we define $\Sigma_s^+ = \mathrm{diag}(\sigma_{s1}^+, \cdots, \sigma_{sn}^+)$ where

$$\sigma_{si}^+ = \begin{cases} 0, & \text{if } 0 \le \sigma_{si} < \bar{\sigma} \\ \sigma_{si}^{-1}, & \text{otherwise} \end{cases} \quad (29)$$

for $i = 1, \cdots, n$, and $\bar{\sigma}$ is a small positive design parameter. According to the above definition, the actual control is obtained as below

$$u = V_s \Sigma_s^+ U_s^{\mathrm{T}} \hat{U} \quad (30)$$

Now, we proceed to develop U_0 in (26) following the backstepping techniques. Firstly, we define the error variables $z_1 = y - y_{\mathrm{d}} = x_1 - y_{\mathrm{d}}$ and $z_2 = x_2 - \alpha_1$, where $\alpha_1 \in \mathbb{R}^n$ is a virtual control variable. Considering system dynamics (5), the time derivative of z_1 is

$$\dot{z}_1 = z_2 + \alpha_1 - \dot{y}_{\mathrm{d}} \quad (31)$$

The virtual control variable is designed as

$$\alpha_1 = \dot{y}_{\mathrm{d}} - L_1 z_1 \quad (32)$$

where $L_1 = L_1^{\mathrm{T}} > 0$. Substituting (32) into (31), we have

$$\dot{z}_1 = z_2 - L_1 z_1 \quad (33)$$

According to (5), the time derivative of z_2 is

$$\dot{z}_2 = \dot{x}_2 - \dot{\alpha}_1 = U + P(x_1, x_2) - \dot{\alpha}_1 \quad (34)$$

where $\dot{\alpha}_1 = \ddot{y}_{\mathrm{d}} - L_1 \dot{z}_1$. To analyze the saturation effect of the estimated nominal control (26), the following auxiliary design system is given

$$\dot{\xi} = \begin{cases} -L_{21}\xi - \frac{|z_2^{\mathrm{T}} L_2 \Delta\hat{U}| + 0.5\Delta\hat{U}^{\mathrm{T}}\Delta\hat{U}}{\|\xi\|^2}\xi + \Delta\hat{U}, & \|\xi\| \ge \chi \\ 0, & \|\xi\| < \chi \end{cases} \quad (35)$$

where $\Delta\hat{U} = \hat{U} - U_0, L_2 = \mathrm{diag}(l_{21}, \cdots, l_{2n}) = L_2^{\mathrm{T}} > 0$, $L_{21} = L_{21}^{\mathrm{T}} > 0$, χ is a small positive design parameter, and $\xi \in \mathbb{R}^n$ is the state of the auxiliary design system. Let $L_{20} = L_{20}^{\mathrm{T}} > 0$, the unsaturated nominal control U_0 is given as

$$\begin{aligned} U_0 = & -L_2^{-1} z_1 - L_2^{-1} L_{20}(z_2 - \xi) \\ & -\{\hat{W}_P\}^{\mathrm{T}} \bullet \{\Phi_P(x_1, x_2)\} + \dot{\alpha}_1 \end{aligned} \quad (36)$$

The term δ and the update laws for vectors $\hat{W}_{Pi}$, $\hat{W}_{Mij}$ in GL matrices $\{\hat{W}_P\}$, $\{\hat{W}_M\}$ are designed as:

$$\delta_{ij} = -\mathrm{sgn}(z_{2i} l_{2i} u_j) s_{ij} \quad (37)$$

$$\dot{\hat{W}}_{Pi} = \Lambda_i(\phi_{Pi}(x_1, x_2) z_{2i} l_{2i} - \beta_i \hat{W}_{Pi}) \quad (38)$$

$$\dot{\hat{W}}_{Mij} = \Gamma_{ij}(\phi_{Mij}(x_1) u_j z_{2i} l_{2i} - \gamma_{ij} \hat{W}_{Mij}) \quad (39)$$

where $\mathrm{sgn}(\cdot)$ is the sign function, s_{ij} is a constant gain that satisfies $s_{ij} \ge \bar{e}_{Mij}$, $\Lambda_i \in \mathbb{R}^{n_{W_P}\times n_{W_P}}, \Lambda_i = \Lambda_i^{\mathrm{T}} > 0$, $\Gamma_{ij} \in \mathbb{R}^{n_{W_M}\times n_{W_M}}, \Gamma_{ij} = \Gamma_{ij}^{\mathrm{T}} > 0$, and $\beta_i, \gamma_{ij} > 0$.

Remark 4: [48] If the use of the discontinuous sign function $\mathrm{sgn}(z_{2i} l_{2i} u_j)$ in (37) is undesired, an alternative δ can be chosen as $\delta_{ij} = -s_{0ij} f(z_{2i} l_{2i} u_j)$, where s_{0ij} is a positive constant and $f(z_{2i} l_{2i} u_j)$ is a continuous odd function of $z_{2i} l_{2i} u_j$. In such a case, it is clear that $z_2^{\mathrm{T}} L_2 (E_M u + \delta u) \le \sum_{i=1}^n \sum_{j=1}^n |z_{2i} l_{2i} u_j| (e_{Mij} - s_{0ij} f(z_{2i} l_{2i} u_j)) \le 0$ for all $z_{2i} l_{2i} u_j$ outside the region

$$\Omega_0 = \{z_{2i} l_2 u_j | \mathrm{sgn}(z_{2i} l_{2i} u_j) f(z_{2i} l_{2i} u_j) < \frac{\bar{e}_{Mij}}{s_{0ij}}\}$$

which can be made arbitrarily small by increasing the value of the design constant s_{0ij}.

Theorem 1: Consider the robotic manipulator (5) satisfying Assumptions 1-2, with the actual control input (30) and neural network weight update laws (38) and (39). For bounded initial conditions, the closed-loop signals z_1, z_2, ξ, $\{\tilde{W}_M\}$ and $\{\tilde{W}_P\}$ are uniformly bounded. In addition, the tracking error z_1 remains within the compact set Ω_{z_1} and it will eventually and exponentially converge to the steady state compact set Ω_s to be defined in the following proof. Finally, motion constraints (6) are not violated.

Proof 1: Consider a Lyapunov function candidate

$$V^* = \frac{1}{2}z_1^{\mathrm{T}}z_1 + \frac{1}{2}\xi^{\mathrm{T}}\xi + \frac{1}{2}z_2^{\mathrm{T}}L_2z_2 \tag{40}$$

In the following derivations, let us first consider the case when $\|\xi\| \geq \chi$ in (35) while the other condition $\|\xi\| < \chi$ will be discussed later. Invoking (33) to (35), we have

$$\begin{array}{rl}\dot{V} & = -z_1^{\mathrm{T}}L_1z_1 + z_1^{\mathrm{T}}z_2 - \xi^{\mathrm{T}}L_{21}\xi - \frac{1}{2}\Delta\hat{U}^{\mathrm{T}}\Delta\hat{U} + \xi^{\mathrm{T}}\Delta\hat{U} \\ & -|z_2^{\mathrm{T}}L_2\Delta\hat{U}| + z_2^{\mathrm{T}}L_2(U + P(x_1,x_2) - \dot{\alpha_1})\end{array} \tag{41}$$

As $-\frac{1}{2}\Delta\hat{U}^{\mathrm{T}}\Delta\hat{U} + \xi^{\mathrm{T}}\Delta\hat{U} \leq \frac{1}{2}\xi^{\mathrm{T}}\xi$, we have

$$\begin{array}{rl}\dot{V} & \leq -z_1^{\mathrm{T}}L_1z_1 + z_1^{\mathrm{T}}z_2 - \xi^{\mathrm{T}}(L_{21} - \frac{1}{2}I)\xi \\ & -|z_2^{\mathrm{T}}L_2\Delta\hat{U}| + z_2^{\mathrm{T}}L_2(U + P(x_1,x_2) - \dot{\alpha_1})\end{array} \tag{42}$$

where I denotes the identity matrix with a proper dimension. As $U = \hat{U} + U - \hat{U} = U_0 + \Delta\hat{U} + U - \hat{U}$, we obtain

$$\begin{array}{rl}\dot{V}^* \leq & -z_1^{\mathrm{T}}L_1z_1 + z_1^{\mathrm{T}}z_2 - \xi^{\mathrm{T}}(L_{21} - \frac{1}{2}I)\xi \\ & +z_2^{\mathrm{T}}L_2(U_0 + U - \hat{U} + P(x_1,x_2) - \dot{\alpha_1})\end{array} \tag{43}$$

Substituting (36) into (43), we have

$$\begin{array}{rl}\dot{V}^* \leq & -z_1^{\mathrm{T}}L_1z_1 - \xi^{\mathrm{T}}(L_{21} - \frac{1}{2}I)\xi - z_2^{\mathrm{T}}L_{20}(z_2 - \xi) \\ & +z_2^{\mathrm{T}}L_2E_P - z_2^{\mathrm{T}}L_2\{\tilde{W}_P\}^{\mathrm{T}} \bullet \{\Phi_P(x_1,x_2)\} \\ & +z_2^{\mathrm{T}}L_2(U - \hat{U})\end{array} \tag{44}$$

where $\{\tilde{W}_P\} = \{\hat{W}_P\} - \{W_P^*\}$. Consider

$$U - \hat{U} = -\{\tilde{W}_M\}^T \bullet \{\Phi_M(x_1)\}u + (E_Mu + \delta u) \tag{45}$$

where $\{\tilde{W}_M\} = \{\hat{W}_M\} - \{W_M^*\}$. Substituting (37) into the term $z_2^{\mathrm{T}}L_2(E_Mu + \delta u)$, we have $z_2^{\mathrm{T}}L_2(E_Mu + \delta u) \leq \sum_{i=1}^n\sum_{j=1}^n |z_{2i}l_{2i}u_j|(e_{Mij} - \mathrm{sgn}(z_{2i}l_{2i}u_j)s_{ij}) \leq 0$.

Through the above mathematical manipulations, we have

$$\begin{array}{rl}\dot{V}^* \leq & -z_1^{\mathrm{T}}L_1z_1 - \xi^{\mathrm{T}}(L_{21} - \frac{1}{2}I)\xi - z_2^{\mathrm{T}}L_{20}(z_2 - \xi) \\ & +z_2^{\mathrm{T}}L_2E_P - z_2^{\mathrm{T}}L_2\{\tilde{W}_P\}^T \bullet \{\Phi_P(x_1,x_2)\} \\ & -z_2^{\mathrm{T}}L_2\{\tilde{W}_M\}^T \bullet \{\Phi(x_1)\}u\end{array} \tag{46}$$

To investigate the convergence of the error signals $\{\tilde{W}_P\}$ and $\{\tilde{W}_M\}$, the augmented Lyapunov function candidate is given as

$$\begin{array}{rl}V & = V^* + \frac{1}{2}\sum_{i=1}^n \tilde{W}_{Pi}^{\mathrm{T}}\Lambda_i^{-1}\tilde{W}_{Pi} \\ & +\frac{1}{2}\sum_{i=1}^n\sum_{j=1}^n \tilde{W}_{Mij}^{\mathrm{T}}\Gamma_{ij}^{-1}\tilde{W}_{Mij}\end{array} \tag{47}$$

It is easy to find that

$$\begin{array}{rl}z_2^{\mathrm{T}}L_{20}\xi & \leq \frac{1}{2}\sigma z_2^{\mathrm{T}}z_2 + \frac{1}{2}\sigma^{-1}\xi^{\mathrm{T}}L_{20}^{\mathrm{T}}L_{20}\xi \qquad (48) \\ z_2^{\mathrm{T}}L_2E_P & \leq \frac{1}{2}z_2^{\mathrm{T}}z_2 + \frac{1}{2}\|L_2E_P\|^2 \qquad (49)\end{array}$$

where $\sigma > 0$. Considering (48), (49) and the update laws (38), (39), we have

$$\begin{array}{rl}\dot{V} & \leq -z_1^{\mathrm{T}}L_1z_1 - \xi^{\mathrm{T}}(L_{21} - 0.5I - 0.5\sigma^{-1}L_{20}^{\mathrm{T}}L_{20})\xi \\ & -z_2^{\mathrm{T}}(L_{20} - (0.5\sigma + 0.5)I)z_2 - \sum_{i=1}^n \beta_i\tilde{W}_{Pi}^{\mathrm{T}}\hat{W}_{Pi} \\ & -\sum_{i=1}^{i=n}\sum_{j=1}^{j=n}\gamma_{ij}\tilde{W}_{Mij}^{\mathrm{T}}\hat{W}_{Mij} + \frac{1}{2}\|L_2E_P\|^2\end{array} \tag{50}$$

Based on the facts

$$\begin{array}{rl}\tilde{W}_{Pi}^{\mathrm{T}}\hat{W}_{Pi} & \geq \frac{1}{2}\|\tilde{W}_{Pi}\|^2 - \frac{1}{2}\|W_{Pi}^*\|^2 \qquad (51) \\ \tilde{W}_{Mij}^{\mathrm{T}}\hat{W}_{Mij} & \geq \frac{1}{2}\|\tilde{W}_{Mij}\|^2 - \frac{1}{2}\|W_{Mij}^*\|^2 \qquad (52)\end{array}$$

we obtain

$$\begin{array}{rl}\dot{V} & \leq -z_1^{\mathrm{T}}L_1z_1 - \xi^{\mathrm{T}}(L_{21} - 0.5I - 0.5\sigma^{-1}L_{20}^{\mathrm{T}}L_{20})\xi \\ & -z_2^{\mathrm{T}}(L_{20} - (0.5\sigma + 0.5)I)z_2 + \frac{1}{2}\|L_2E_P\|^2 \\ & -\sum_{i=1}^n \frac{\beta_i}{2}\|\tilde{W}_{Pi}\|^2 - \sum_{i=1}^{i=n}\sum_{j=1}^{j=n}\frac{\gamma_{ij}}{2}\|\tilde{W}_{Mij}\|^2 \\ & +\sum_{i=1}^n \frac{\beta_i}{2}\|W_{Pi}^*\|^2 + \sum_{i=1}^{i=n}\sum_{j=1}^{j=n}\frac{\gamma_{ij}}{2}\|W_{Mij}^*\|^2 \\ & \leq -\rho V + \zeta \qquad (53) \\ \rho & = \min\big(2\lambda_{\min}(L_1), 2\lambda_{\min}(\bar{L}_{21}), 2\lambda_{\min}(\bar{L}_{20}L_2^{-1}), \\ & \min(\frac{\beta_i}{\lambda_{\max}(\Lambda_i^{-1})}), \min(\frac{\gamma_{ij}}{\lambda_{\max}(\Gamma_{ij}^{-1})})\big) \qquad (54) \\ \zeta & = \sum_{i=1}^n \frac{\beta_i}{2}\|W_{Pi}^*\|^2 + \sum_{i=1}^n\sum_{j=1}^n\frac{\gamma_{ij}}{2}\|W_{Mij}^*\|^2 + \frac{1}{2}\|L_2E_P\|^2 \qquad (55)\end{array}$$

with $\bar{L}_{20} = L_{20} - (0.5\sigma + 0.5)I$ and $\bar{L}_{21} = L_{21} - 0.5I - 0.5\sigma^{-1}L_{20}^{\mathrm{T}}L_{20}$. To ensure $\rho > 0$, the design parameters $\beta_i > 0, \gamma_{ij} > 0, \sigma > 0, L_2 = L_2^{\mathrm{T}} > 0, L_{20} = L_{20}^{\mathrm{T}} > 0$ and $L_{21} = L_{21}^{\mathrm{T}} > 0$ are to satisfy the following conditions

$$\lambda_{\min}(L_1) > 0,\ \lambda_{\min}(\bar{L}_{21}) > 0,\ \lambda_{\min}(\bar{L}_{20}) > 0 \tag{56}$$

According to Lemma 1.2 i) in [49], (53) indicates that all the closed-loop signals z_1, z_2, ξ, $\{\tilde{W}_M\}$ and $\{\tilde{W}_P\}$ are uniformly bounded. Particularly, the tracking error z_1 is uniformly bounded by a compact set Ω_{z_1} derived as follows:

Multiplying (53) by $e^{\rho t}$ yields

$$\frac{\mathrm{d}}{\mathrm{d}t}(V(t)e^{\rho t}) \leq \zeta e^{\rho t}. \tag{57}$$

By integrating (57) over $[0, t]$, we obtain

$$0 \leq V(t) \leq \big(V(0) - \frac{\zeta}{\rho}\big)e^{-\rho t} + \frac{\zeta}{\rho} \tag{58}$$

From (58), it is easy to find that

$$0 \leq V(t) \leq \big(V(0) - \frac{\zeta}{\rho}\big)e^{-\rho t} + \frac{\zeta}{\rho} \leq V(0) + \frac{\zeta}{\rho}. \tag{59}$$

According to (47), we have $\frac{1}{2}z_1^{\mathrm{T}}z_1 \leq V(t)$. Thus,

$$\|z_1(t)\|^2 \leq 2(V(0) + \frac{\zeta}{\rho}). \tag{60}$$

Hence, the tracking error z_1 is uniformly bounded by the compact set

$$\Omega_{z_1} = \{z_1(t) \| \|z_1(t)\| \leq \sqrt{2(V(0) + \frac{\zeta}{\rho})}\} \tag{61}$$

In addition, according to (59) and Lemma 1.2 ii) in [49], we can see that z_1 is uniformly ultimately bounded and it

exponentially converges to the following steady state compact set

$$\Omega_s = \{x(t) | \lim_{t\to\infty} \|z_1(t)\| = \sqrt{\frac{2\zeta}{\rho}}\} \tag{62}$$

From (54) and (55), the size of Ω_s can be reduced to be sufficiently small by designing ρ such that it is large enough. In addition, from (58), the convergence speed of z_1 can also be boosted with a large ρ. In other words, the robot's end-effector can be kept inside the constrained region at steady state by simply designing ρ that is large enough.

Remark 5: The state ξ in the auxiliary design system (35) indicates whether there exists saturation of the nominal input. In particular, $\|\xi\| \geq \chi$ means that there exists saturation of U_0, which has been considered in the above stability analysis. If there is no saturation, we have $\|\xi\| < \chi$ and $\Delta\hat{U} = 0$, i.e., $\hat{U} = U_0$, and thus the nominal input is bounded. Therefore, following (40) to (55), it is easy to show that **Theorem** 1 is still valid [38].

Remark 6: To guarantee that the approximation property of RBFNNs (20) and (21) holds, their inputs have to be in the corresponding compact sets, i.e., $x_1 \in \Omega_{x_1}$ and $[x_1^{\mathrm{T}}, x_2^{\mathrm{T}}]^{\mathrm{T}} \in \Omega_{x_1,x_2}$, respectively. From **Theorem** 1 and **Proof** 1, it is clear that the ultimate bound of z_1 can be made arbitrarily small by properly choosing the parameters such as L_1 and L_{20}. Therefore, x_1 and $[x_1^{\mathrm{T}}, x_2^{\mathrm{T}}]^{\mathrm{T}}$ can be guaranteed to be in the corresponding compact sets.

Remark 7: In **Theorem** 1,the designed unsaturated nominal control U_0 and the corresponding weight update laws (38) and (39) lead to the Uniformly Ultimate Boundedness (UUB) of the tracking error z_1. It is shown in **Proof** 1 that this ultimate bounds can be made arbitrarily small by increasing ρ. It is worth mentioning that the asymptotic convergence result of z_1 can also be obtained within our control design framework. To do this, we can simply add a term $-\mathbf{sgn}(z_2)z_2L_2\bar{E}_p$ to U_0 in (36) to cancel the effects of the NN approximation error E_P in Lyapunov analysis (46) and choose $\gamma_{ij} = \beta_i = 0$. In this way, it is clear that $\dot{V} \leq 0$ and $\dot{V} < 0$ for all $z_1 \neq 0$. However, this result comes at a cost of possible chattering in the system due to usage of the discontinuous sign function.

Remark 8: The proposed method is motivated by controlling a robotic manipulator with motion constraints, but it is also applicable to other systems which are affine in control, e.g., ocean vessels [50].

IV. Simulation

In this section, we simulate a 2-DOFs robotic manipulator moving in a horizontal plane under different types of motion constraints. The following general simulation scenarios can reflect a variety of new robotic applications such as robotic tool use [51] and human-robot physical collaboration [52], especially in unstructured environments where asymmetric, time-varying and task-dependent motions constraints are usually required to guarantee the safety of operations. The dynamic model of the robotic manipulator is given by Eq. (63) and values of the parameters are given in Table I.

TABLE I
Parameters of the robotic manipulator

Parameter	Description	Value
m_1	Mass of link 1	1.1510kg
m_2	Mass of link 2	0.5755kg
$l_1 = 2l_{c1}$	Length of link 1	0.31m
$l_2 = 2l_{c2}$	Length of link 2	0.34m
I_1	Moment of inertia of link 1	0.3kgm^2
I_2	Moment of inertia of link 2	0.3kgm^2

The initial positions of the end-effector are given as $q(0) = [-\frac{1}{2}\pi, \frac{1}{2}\pi]^{\mathrm{T}}$. The desired trajectory is a circular path in the task space, which is described by $y_{d1}(t) = 0.5\cos(0.5t)$ and $y_{d2}(t) = 0.5\sin(0.5t)$. To quantify and evaluate the tracking performance, the tracking error is defined as $e_i = |y_i - y_{\mathrm{d}i}|$ for $i = 1, 2$. Motion constraints of the robotic manipulator are defined as $-0.6 \leq y_1 \leq 0.6, -0.6 \leq y_2 \leq 0.6, -0.4 \leq \dot{y}_1 \leq 0.4$ and $-0.4 \leq \dot{y}_2 \leq 0.4$.

The design parameters are chosen as $k_1 = [1000, 1000]^{\mathrm{T}}$, $k_2 = [100, 100]^{\mathrm{T}}$, $k_3 = [10^4, 10^4]^{\mathrm{T}}$, $L_1 = \mathrm{diag}[5, 7]$, $L_{20} = \mathrm{diag}[500, 500]$, $L_{21} = \mathrm{diag}[1000, 1000]$, $L_2 = \mathrm{diag}[1, 2]$, $\sigma = 200$, $\chi = 0.001$, $s_{i,j} = 0.001$ for $i, j = 1, 2$, $\xi(0) = [0.001, 0.001]^{\mathrm{T}}$ and $\bar{\sigma} = 10^{-10}$. For the RBFNNs (22) and (23), the number of nodes $n_{w_M} = n_{w_P} = 2^7 = 128$. $\Gamma_{i,j} = \Lambda_i = 0.01 \times I_{128\times128}$ and $\beta_i = \gamma_{i,j} = 0.1$ for $i, j = 1, 2$. The centers of the radial basis functions are evenly distributed in $[-1, 1]$ and their variance is set to be 1. The initial weights are $w_{M11} = 0.3 \times [1, 1, \cdots, 1]^{\mathrm{T}} \in \mathbb{R}^{128}$ and $w_{M12} = w_{M21} = w_{M22} = w_{P1} = w_{P2} = 0.1 \times [1, 1, \cdots, 1]^{\mathrm{T}} \in \mathbb{R}^{128}$.

A guideline for choosing the above parameters are given as follows: as the parameters k_1, k_2, k_3 are used when converting a motion constraint into the constraint of its derivative in (9), (11) and (12), they should be sufficiently large so that the converted motion constraints are hard enough and the resulted constrained region will not be over-conservative. The diagonal matrices L_1, L_{20}, L_2 are gains for the signals z_1 and z_2 used in the standard back-stepping control design procedure. Generally, L_1 and L_{20} should be tuned according to the position and velocity errors, respectively and L_2 is a global scaling factor for these two gains. L_{21} is a gain for the auxiliary signal designed to handle the saturation effects of the nominal input signals, which generally gives satisfactory results with the default values. As mentioned in **Proof 1**, all these gains should be properly chosen such that (56) is satisfied. On the other hand, parameters $\sigma, s, \chi, \bar{\sigma}$ are used for stability analysis and there is generally no need for tuning them. As for the learning rates $\Lambda, \Gamma, \lambda, \gamma$, they should be chosen to be large enough to achieve fast and stable online learning. More instructions on designing adaptive RBFNNs can be found in [39] and other related papers.

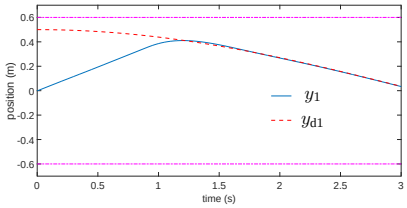
To investigate the efficacy of the proposed method, comparative simulation studies are conducted on its transient responses with and without motion constraints and the results are shown in Figs. 1 and 2. Note that the adaptive control without motion constraints in Fig. 2 is also implemented with the proposed method, where position and velocity constraints are defined with infinite values.

Figure 1 shows that, under the proposed control, the end-effector of the robot successfully tracks the desired trajectory

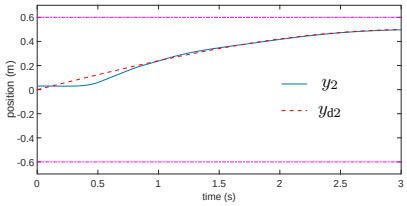
$$M(q) = \begin{bmatrix} m_1 l_{c1}^2 + m_2(l_1^2 + l_{c2}^2 + 2l_1 l_{c2}\cos q_2) + I_1 + I_2 & m_2(l_{c2}^2 + l_1 l_{c2}\cos q_2) + I_2 \\ m_2(l_{c2}^2 + l_1 l_{c2}\cos q_2) + I_2 & m_2 l_{c2}^2 + I_2 \end{bmatrix},$$

$$C(q,\dot{q}) = \begin{bmatrix} -m_2 l_1 l_{c2}\dot{q}_2 \sin q_2 & -m_2 l_1 l_{c2}(\dot{q}_1 + \dot{q}_2)\sin q_2 \\ m_2 l_1 l_{c2}\dot{q}_1 \sin q_2 & 0 \end{bmatrix}, \; G(q) = 0 \tag{63}$$

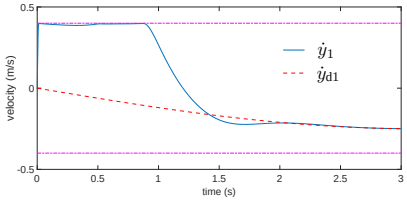
while not violating the motion constraints. During the transient response, the position and velocity of the end-effector are perfectly limited within the constrained region marked by the pink dash-dotted lines. More specifically, in Figs. 1(c) and 1(d), we can observe that the end-effector's velocity stops increasing when it reaches the boundary of the constrained region. This demonstrates that the proposed control can effectively prevent the violation of motion constraints.



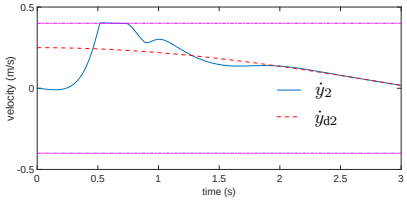
(a) y_1 and y_{d1}



(b) y_2 and y_{d2}



(c) $\dot{y}_1$ and $\dot{y}_{d1}$

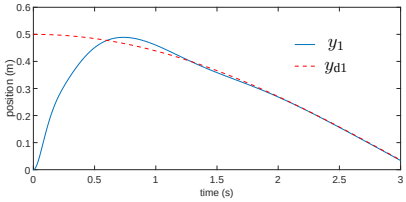


(d) $\dot{y}_2$ and $\dot{y}_{d2}$

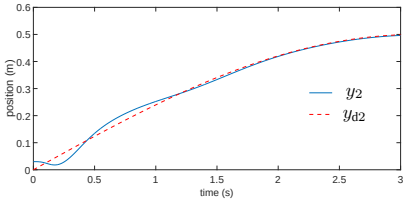
Fig. 1. Robot's transient response with motion constraints

A more illustrative demonstration of the efficacy of the proposed method can be obtained by comparing above results with those in Fig. 2, where no motion constraint is imposed and the simulated situation is reduced to a standard adaptive tracking control problem. As can be seen from Fig. 2, the end-effector also successfully tracks the desired trajectory quickly. However, since no motion constraint is considered, the velocities of the end-effector rise to a much higher level compared to those in Figs. 1(c) and 1(d) as the system driven by the designed control tries to track the desired trajectory as soon as possible.

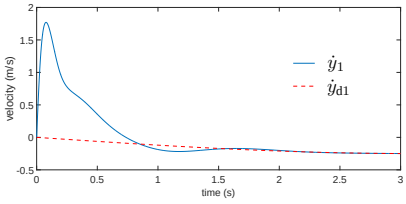
From this comparison, we show that motion constraints can be incorporated into our control design without causing any degradation to the tracking performance. In addition, for applications where the overshoot of position and/or velocity in the transient response is undesired, the proposed control provides an explicit and direct way to address this problem by adding some reasonable motion constraints into consideration.



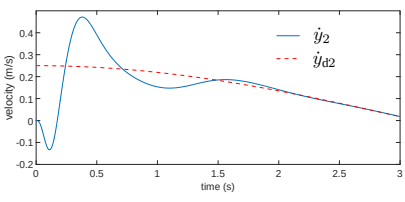
(a) y_1 and y_{d1}



(b) y_2 and y_{d2}



(c) $\dot{y}_1$ and $\dot{y}_{d1}$

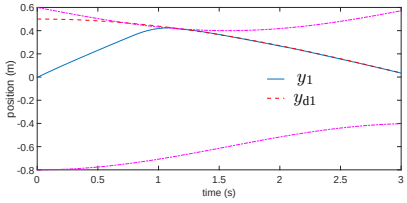


(d) $\dot{y}_2$ and $\dot{y}_{d2}$

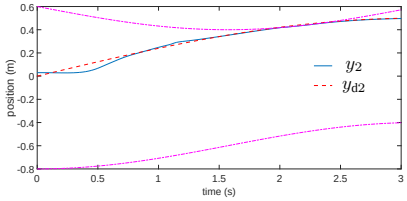
Fig. 2. Robot's transient response without motion constraints

To further investigate the efficacy of the proposed control, we consider another simulation where the motion constraints are asymmetric and time-varying. Specifically, we require $-0.6 - 0.2\cos(t) \leq y_1(t) \leq 0.6 - 0.2\sin(t), -0.6 -$

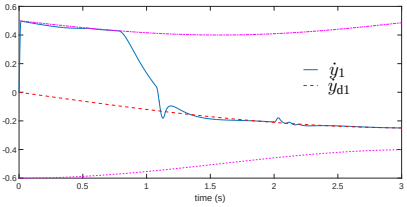
$0.2\cos(t) \leq y_2(t) \leq 0.6 - 0.2\sin(t), -0.5 - 0.1\cos(t) \leq \dot{y}_1(t) \leq 0.5 - 0.1\sin(t)$ and $-0.5 - 0.1\cos(t) \leq \dot{y}_2(t) \leq 0.5 - 0.1\sin(t)$. The simulation results are shown in Fig. 3



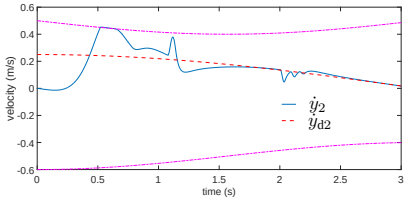
(a) y_1 and y_{d1}



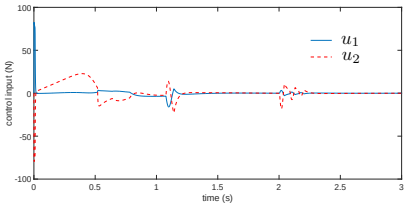
(b) y_2 and y_{d2}



(c) $\dot{y}_1$ and $\dot{y}_{d1}$



(d) $\dot{y}_2$ and $\dot{y}_{d2}$



(e) Control input u

Fig. 3. Robot's transient response with time-varying asymmetric motion constraints

From Fig. 3, we can observe that excellent tracking performance can still be achieved with the same configuration of control parameters when the motion constraints have varied significantly. Both of the time-varying asymmetric position and velocity constraints (marked by the pink dash-dotted curves) are not violated even when the constrained regions are shrinking at the beginning of the transient response. These results show the efficacy of the proposed method in handling time-varying motion constraints, which corresponds to the cases when motion constraints are a function of time, the robot's state or its operation environment.

It is worth mentioning that larger design parameters k_1, k_2 and k_3 result in harder motion constraints, i.e., the position/velocity will get nearer to or even stay at the boundaries of the constrained region when it is to escape this region. However, a larger control input is required to keep it at the boundary, as shown in Fig. 3(e), and mild chattering in the control input may appear. Therefore, a trade-off between tracking performance at the boundaries of the constrained region and control effort needs to be considered.

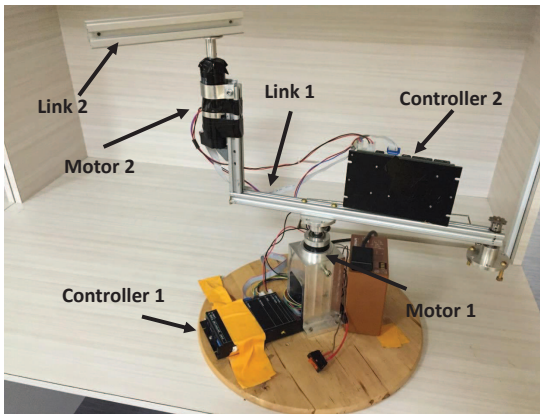


Fig. 4. The 2-DOF planar robotic manipulator

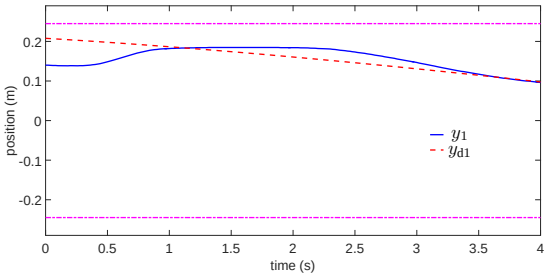
V. Experiment

In this section, the proposed method is further examined through an experiment, which is conducted on a 2-DOF planar robotic manipulator as shown in Fig. 4. The robot is composed of 2 MAXON motors with two MAXON EPOS2 70/10 dual loop controllers. The links of the robot are of lengths $l_1 = 0.14$m and $l_2 = 0.15$m. A desktop PC is used to process the collected data and implement the proposed method.

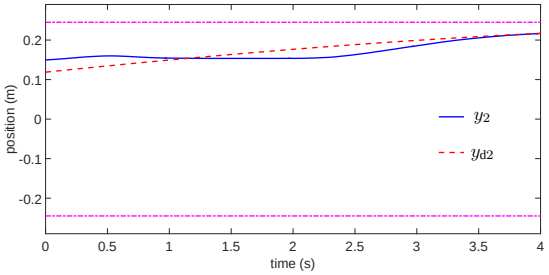
The initial joint position of the robot is $q(0) = [0, \frac{\pi}{2}]^{\mathrm{T}}$ and the desired trajectory of the end-effector is $y_{\mathrm{d}}(t) = [0.24\cos(0.05\pi t + \frac{\pi}{6}), 0.24\sin(0.05\pi t + \frac{\pi}{6})]^{\mathrm{T}}$ for $t \in [0, 4]$s. The motion constraints are defined as $-0.245 \leq y_1 \leq 0.245, -0.245 \leq y_2 \leq 0.245, -0.1 \leq \dot{y}_1 \leq 0.1$ and $-0.05 \leq \dot{y}_2 \leq 0.05$.

The design parameters are chosen as $k_1 = [1, 30]^{\mathrm{T}}$, $k_2 = [0.05, 1]^{\mathrm{T}}$, $k_3 = [1, 1]^{\mathrm{T}}$, $L_1 = \mathrm{diag}(30, 30)$, $L_{20} = \mathrm{diag}(130, 12)$, $L_{21} = \mathrm{diag}(500, 500)$, $L_2 = \mathrm{diag}(2.5, 0.4)$, $\sigma = 20$, $\chi = 0.001$, $s_{i,j} = 0.001$ for $i, j = 1, 2$, $\xi(0) = [0.001, 0.001]^{\mathrm{T}}$ and $\bar{\sigma} = 10^{-10}$. For the RBFNNs (22) and (23), the number of nodes $n_{W_M} = n_{W_P} = 2^7 = 128$. The centers of the radial basis functions are evenly distributed in $[-1, 1]$ and their variance is set to be 1. $\Lambda_i = 0.1 I_{128\times128}, \beta_i = 0.1$, $\Gamma_{1,1} = \Gamma_{1,2} = \Gamma_{2,1} = 0.05 I_{128\times128}$, $\Gamma_{2,2} = 0.005 I_{128\times128}$, and $\gamma_{i,j} = 0.01$ for $i, j = 1, 2$. The initial weights are $W_{M11} = 0.6 \times [1, 1, \cdots, 1]^{\mathrm{T}} \in \mathbb{R}^{128}$, $W_{M12} = W_{M21} = W_{M22} = -0.1 \times [1, 1, \cdots, 1]^{\mathrm{T}} \in \mathbb{R}^{128}$, and $W_{P1} = W_{P2} = 0.1 \times [1, 1, \cdots, 1]^{\mathrm{T}} \in \mathbb{R}^{128}$.

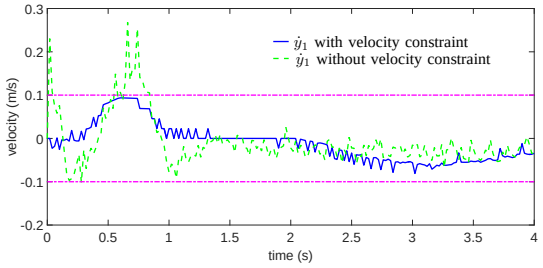
The transient response of the robot is shown in Fig. 5. As shown in Figs. 5(a) and 5(b), the end-effector is able to track the desired trajectory in about 3.5 seconds and position constraints (marked by pink dash-dotted lines) are constantly satisfied. From Figs. 5(c) and 5(d), we can see that the velocity



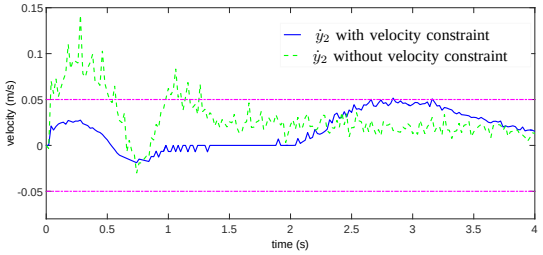
(a) y_1 and y_{d1}



(b) y_2 and y_{d2}



(c) $\dot{y}_1$



(d) $\dot{y}_2$

Fig. 5. Robot's transient response with motion constraints in experiments

of the end-effector is successfully limited within the prescribed constrained region. For the comparison purpose, we consider another experiment without velocity constraints. As a result, a larger velocity appears in the beginning of the transient response, as also shown in Figs. 5(c) and 5(d), due to the fact that the control tries to drive the end-effector to the desired position as fast as possible. These results are in accordance to the simulation results in the previous section, and illustrate the efficacy of the proposed method. Different from the simulation results, the velocities in the experiment are not so smooth, which is because they are obtained by differentiating the positions collected from the encoders. A filter may be designed to obtain a cleaner velocity signal if it is required in some applications.

VI. Conclusion

In this paper, we have investigated the control of a robotic manipulator with parametric uncertainties and two types of motion constraints. A framework has been proposed to convert motion constraints of different types into a unified constraint of the nominal input, based on which an adaptive neural network control has been developed. The efficacy of the proposed method has been demonstrated through simulation and experiment studies.

References

[1] Y. Li and S. S. Ge, "Human-robot collaboration based on motion intention estimation," *IEEE/ASME Trans. Mechatronics*, vol. 19, no. 3, pp. 1007–1014, 2014.
[2] N. Najmaei and M. R. Kermani, "Applications of artificial intelligence in safe human–robot interactions," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 41, no. 2, pp. 448–459, 2011.
[3] G. Du, P. Zhang, and D. Li, "Human–manipulator interface based on multisensory process via kalman filters," *IEEE Trans. Ind. Electron.*, vol. 61, no. 10, pp. 5411–5418, 2014.
[4] G. Du and P. Zhang, "A markerless human–robot interface using particle filter and kalman filter for dual robots," *IEEE Trans. Ind. Electron.*, vol. 62, no. 4, pp. 2257–2264, 2015.
[5] W. He, S. S. Ge, Y. Li, E. Chew, and Y. S. Ng, "Neural network control of a rehabilitation robot by state and output feedback," *J. Intel. Robot. Syst.*, vol. 80, no. 1, pp. 15–31, 2015.
[6] J. Funda, R. H. Taylor, B. Eldridge, S. Gomory, and K. G. Gruben, "Constrained cartesian motion control for teleoperated surgical robots," *IEEE Trans. Robot. Autom.*, vol. 12, no. 3, pp. 453–465, 1996.
[7] C. O. A. S. T. W. . G. H. A. Albu-Schäffer, S. Haddadin, "The dlr lightweight robot: design and control concepts for robots in human environments," *Ind. Robot*, vol. 34, no. 5, pp. 376–385, 2007.
[8] K. B. Ngo and R. Mahony, "Bounded torque control for robot manipulators subject to joint velocity constraints," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2006, pp. 7–12.
[9] X. Papageorgiou and K. J. Kyriakopoulos, "Motion tasks for robot manipulators subject to joint velocity constraints," in *Proc. IEEE Int. Conf. Int. Robot. Syst. (IROS)*, 2008, pp. 2139–2144.
[10] Y. Zhang, S. S. Ge, and T. H. Lee, "A unified quadratic-programming-based dynamical system approach to joint torque optimization of physically constrained redundant manipulators," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 34, no. 5, pp. 2126–2132, 2004.
[11] K. P. Tee, R. Yan, and H. Li, "Adaptive admittance control of a robot manipulator under task space constraint," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2010, pp. 5181–5186.
[12] W. He, A. O. David, Z. Yin, and C. Sun, "Neural network control of a robotic manipulator with input deadzone and output constraint," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 46, no. 6, pp. 759–770, 2016.
[13] S. G. Khan, G. Herrmann, F. L. Lewis, T. Pipe, and C. Melhuish, "A novel q-learning based adaptive optimal controller implementation for a humanoid robotic arm," in *Proc. 18th IFAC World Congr.*, 2011, no. 1, pp. 13 528–13 533.
[14] C.-C. Cheah, D. Q. Wang, and Y. C. Sun, "Region-reaching control of robots," *IEEE Trans. Robot.*, vol. 23, no. 6, pp. 1260–1264, 2007.
[15] C. C. Cheah, S. P. Hou, and J. J. E. Slotine, "Region-based shape control for a swarm of robots," *Automatica*, vol. 45, no. 10, pp. 2406–2411, 2009.
[16] S. P. Hou and C. C. Cheah, "Can a simple control scheme work for a formation control of multiple autonomous underwater vehicles?" *IEEE Trans. Control Syst. Technol.*, vol. 19, no. 5, pp. 1090–1101, 2011.
[17] X. Li and C. C. Cheah, "Adaptive neural network control of robot based on a unified objective bound," *IEEE Trans. Control Syst. Technol.*, vol. 22, no. 3, pp. 1032–1043, 2014.
[18] J. E. Slotine and W. Li, "On the adaptive control of robot manipulators," *Int. J. Robot. Res.*, vol. 6, no. 3, pp. 49–59, 1987.
[19] S. Riverso and G. Ferrari-Trecate, "Tube-based distributed control of linear constrained systems," *Automatica*, vol. 48, no. 11, pp. 2860–2865, 2012.
[20] A. Kojima and M. Morari, "Lq control for constrained continuous-time systems," *Automatica*, vol. 40, no. 7, pp. 1143–1155, 2004.
[21] X. Jin and J. X. Xu, "Iterative learning control for output-constrained systems with both parametric and nonparametric uncertainties," *Automatica*, vol. 49, no. 8, pp. 2508–2516, 2013.

[22] S. Piche, B. Sayyar-Rodsari, D. Johnson, and M. Gerules, "Nonlinear model predictive control using neural networks," *IEEE Control Syst. Mag.*, vol. 20, no. 3, pp. 53–62, 2000.
[23] Z. Yan and J. Wang, "Robust model predictive control of nonlinear systems with unmodeled dynamics and bounded uncertainties based on neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 3, pp. 457–469, 2014.
[24] G. P. Liu and V. Kadirkamanathan, "Predictive control for non-linear systems using neural networks," *Int. J. Control*, vol. 71, no. 6, pp. 1119–1132, 1998.
[25] J. Huang and F. L. Lewis, "Neural-network predictive control for nonlinear dynamic systems with time-delay," *IEEE Trans. Neural Netw.*, vol. 14, no. 2, pp. 377–389, 2003.
[26] S. S. Ge, C. Yang, and T. H. Lee, "Adaptive predictive control using neural network for a class of pure-feedback systems in discrete time," *IEEE Trans. Neural Netw.*, vol. 19, no. 9, pp. 1599–1614, 2008.
[27] W. He, C. Sun, and S. S. Ge, "Top tension control of a flexible marine riser by using integral-barrier lyapunov function," *IEEE/ASME Trans. Mechatronics*, vol. 20, no. 2, pp. 497–505, 2015.
[28] Z.-L. Tang, S. S. Ge, K. P. Tee, and W. He, "Robust adaptive neural tracking control for a class of perturbed uncertain nonlinear systems with state constraints," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. PP, no. 99, pp. 1–12, 2016.
[29] Y. J. Liu and S. Tong, "Barrier lyapunov functions-based adaptive control for a class of nonlinear pure-feedback systems with full state constraints," *Automatica*, vol. 64, pp. 70–75, 2016.
[30] Y. H. Kim and F. L. Lewis, "Neural network output feedback control of robot manipulators," *IEEE Trans. Robot. Autom.*, vol. 15, no. 2, pp. 301–309, 1999.
[31] R. R. Selmic and F. L. Lewis, "Deadzone compensation in motion control systems using neural networks," *IEEE Trans. Autom. Control*, vol. 45, no. 4, pp. 602–613, 2000.
[32] Z. Liu, F. Wang, and Y. Zhang, "Adaptive visual tracking control for manipulator with actuator fuzzy dead-zone constraint and unmodeled dynamic," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 45, no. 10, pp. 1301–1312, 2015.
[33] Y. J. Liu and S. Tong, "Adaptive fuzzy control for a class of unknown nonlinear dynamical systems," *Fuzzy Sets Syst.*, vol. 263, pp. 49–70, 2015.
[34] W. He, Y. Dong, and C. Sun, "Adaptive neural impedance control of a robotic manipulator with input saturation," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 46, no. 3, pp. 334–344, 2016.
[35] X. Jin, "Adaptive fault tolerant control for a class of input and state constrained mimo nonlinear systems," *In. J. Robust Nonlinear Control*, vol. 26, no. 2, pp. 286–302, 2016.
[36] C. P. Bechlioulis and G. A. Rovithakis, "Prescribed performance adaptive control for multi-input multi-output affine in the control nonlinear systems," *IEEE Trans. Autom. Control*, vol. 55, no. 5, pp. 1220–1226, 2010.
[37] S. I. Han and J. M. Lee, "Output-tracking-error-constrained robust positioning control for a nonsmooth nonlinear dynamic system," *IEEE Trans. Ind. Electron.*, vol. 61, no. 12, pp. 6882–6891, 2014.
[38] M. Chen, S. S. Ge, and B. Ren, "Adaptive tracking control of uncertain mimo nonlinear systems with input constraints," *Automatica*, vol. 47, no. 3, pp. 452–465, 2011.
[39] S. S. Ge, T. H. Lee, and C. J. Harris, *Adaptive Neural Network Control of Robotic Manipulators*. London, U.K.: World Scientific, 1998.
[40] J. Campos and F. L. Lewis, "Deadzone compensation in discrete time using adaptive fuzzy logic," *IEEE Trans. Fuzzy Syst.*, vol. 7, no. 6, pp. 697–707, 1999.
[41] S. Haykin, *Neural networks: a comprehensive foundation*. New Jersey, USA: Prentice Hall, 1994.
[42] S. S. Ge and C. Wang, "Adaptive nn control of uncertain nonlinear pure-feedback systems," *Automatica*, vol. 38, no. 4, pp. 671–682, 2002.
[43] Y. Gao and Y. J. Liu, "Adaptive fuzzy optimal control using direct heuristic dynamic programming for chaotic discrete-time system," *J. Vib. Control*, vol. 22, no. 2, pp. 595–603, 2016.
[44] Y. J. Liu, S. Tong, D.-J. Li, and Y. Gao, "Fuzzy adaptive control with state observer for a class of nonlinear discrete-time systems with input constraint," *IEEE Trans. Fuzzy Syst.*, vol. PP, no. 99, pp. 1–1, 2015.
[45] Y. J. Liu, Y. Gao, S. Tong, and Y. Li, "Fuzzy approximation-based adaptive backstepping optimal control for a class of nonlinear discrete-time systems with dead-zone," *IEEE Trans. Fuzzy Syst.*, vol. 24, no. 1, pp. 16–28, 2016.
[46] J.-J. E. Slotine, W. Li *et al.*, *Applied nonlinear control*. Prentice-Hall Englewood Cliffs, NJ, 1991, vol. 199, no. 1.
[47] G. H. Golub and C. Reinsch, "Singular value decomposition and least squares solutions," *Numerische Mathematik*, vol. 14, no. 5, pp. 403–420, 1970.
[48] M. Polycarpou, J. Farrell, and M. Sharma, "On-line approximation control of uncertain nonlinear systems: issues with control input saturation," in *Proc. Am. Control Conf.*, 2003, pp. 543–548.
[49] S. S. Ge and C. Wang, "Adaptive neural control of uncertain mimo nonlinear systems," *IEEE Trans. Neural Netw.*, vol. 15, no. 3, pp. 674–692, 2004.
[50] K. P. Tee and S. S. Ge, "Control of fully actuated ocean surface vessels using a class of feedforward approximators," *IEEE Trans. Control Syst. Technol.*, vol. 14, no. 4, pp. 750–756, 2006.
[51] C. Eppner, R. Deimel, J. Álvarez-Ruiz, M. Maertens, and O. Brock, "Exploitation of environmental constraints in human and robotic grasping," *Int. J. Robot. Res.*, vol. 34, no. 7, pp. 1021–1038, 2015.
[52] Y. Li, K. P. Tee, W. L. Chan, R. Yan, Y. Chua, and D. K. Limbu, "Continuous role adaptation for human–robot shared control," *IEEE Trans. Robot.*, vol. 31, no. 3, pp. 672–681, 2015.

Mingming Li Mingming Li received the B.Eng. degree in communication engineering from Sun Yat-sen University, Guangzhou, China, in 2013. He is currently pursuing his Ph.D. degree with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore. His current research interests include adaptive control, associative memories and artificial intelligence.

Yanan Li (M'14) received the B.Eng degree in control science and engineering and the M.Eng degree in control and mechatronics engineering, from the Harbin Institute of Technology, China, in 2006 and 2009, respectively, and the Ph.D. degree from the NUS Graduate School for Integrative Sciences and Engineering, National University of Singapore, Singapore, in 2013. He has been a Research Scientist at the Institute for Infocomm Research, Agency for Science, Technology and Research (A*STAR), Singapore from 2013 to 2015. Currently, he is a Research Associate at the Department of Bioengineering, Imperial College London, UK. His general research interests include physical human-robot interaction and human-robot collaboration.

Shuzhi Sam Ge (S'90-M'92-SM'99-F'06) received the B.Sc. degree from Beijing University of Aeronautics and Astronautics, Beijing, China, in 1986, and the Ph.D. degree from the Imperial College of Science, Technology and Medicine, University of London, London, U.K., in 1993.

He is the Founding Director of the Social Robotics Laboratory, Interactive Digital Media Institute, National University of Singapore and on leave with University of Electronic Science and Technology of China, Chengdu 610054, China. He has authored or coauthored six books and more than 400 international journal and conference papers. His current research interests include social robotics, multimedia fusion, medical robots, and intelligent systems.

Dr. Ge is the Editor-in-Chief of the International Journal of Social Robotics. He has served/been serving as an Associate Editor for a number of flagship journals. He also serves as an Editor of the Taylor & Francis Automation and Control Engineering Series. He also served as the Vice President of Technical Activities, 2009-2010, and the Vice President for Membership Activities, 2011-2012, IEEE Control Systems Society.

Tong Heng Lee (M'90) received the B.A. degree (with first class honors) in engineering tripos from Cambridge University, Cambridge, U.K., in 1980 and the Ph.D. degree in electrical engineering from Yale University, New Haven, CT, in 1987.

He is a Professor with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore, and is currently the Head of the Drives, Power, and Control Systems Group in this department. He is the Deputy Editor-in-Chief of the International Federation of Automatic Control Mechatronics International Journal and serves as the Associate Editor of many other flagship journals. He has also coauthored three research monographs and holds four patents (two of which are in the technology area of adaptive systems, and the other two are in the area of intelligent mechatronics). His research interests are in the areas of adaptive systems, knowledge-based control, intelligent mechatronics, and computational intelligence.

Dr. Lee received the Cambridge University Charles Baker Prize in Engineering.